

CLAIMS

What is claimed is:

1. A system for facilitating adaptive grid-based document layout, the system comprising:
 - 5 a template storage unit adapted to store a plurality of templates;
 - a layout engine adapted to apply document content to a template of said plurality of templates, wherein said layout engine is further adapted to determine a score providing a measure of how well said document content fits said template; and
 - 10 a paginator adapted to provide said document content and said plurality of templates to said layout engine, receive said quality score from said layout engine, and determine a desirable pairing of document content and said plurality of templates.
2. The system of claim 1, wherein the paginator is further adapted to select a set of templates from said plurality of templates and to provide said set of templates to said layout engine for processing.
 - 15 3. The system of claim 1, wherein said quality score determined by said layout engine is an optimization problem which seeks to optimize a second quality score across interdependent elements within said template.
- 20 4. The system of claim 1, wherein the system further comprises:
 - a style sheets storage unit adapted to store a plurality of style sheets, wherein said plurality of style sheets specify the styling of text in said document content.
- 25 5. The system of claim 1, wherein the system further comprises:
 - a template authoring tool adapted to create and modify templates of said plurality of templates; and
 - a user interface adapted to receive output from said template authoring tool and display said output to a user.

30

6. The system of claim 5, wherein one template of said plurality of templates comprises a parent template having characteristics, and wherein said template authoring tool is further adapted to create and modify a child template that inherits characteristics from said parent template.

5

7. The system of claim 6, wherein said characteristics include guidelines, elements, constraints, and preconditions.

8. The system of claim 5, wherein said template authoring tool is further adapted to assign a fitness scoring function to a template of said plurality of templates, wherein said fitness scoring function is customizable for determining a fitness of said template for a portion of said document content.

10 9. The system of claim 8, wherein said user interface is further adapted to assign an attribute preference to an element within said template, wherein said attribute preference is used to determine said fitness of said template.

15 10. The system of claim 9, wherein said attribute preference comprises a first attribute preference of a plurality of attribute preferences, and wherein said user interface is further adapted to order a subset of said attribute preferences in a sequence defining the importance of said attribute preferences to said layout engine.

20 11. The system of claim 5, wherein said user interface is further adapted to organize said set of templates in a recursive template configuration.

25

12. The system of claim 11, wherein said recursive template configuration permits a first template to be nested within a second template.

30 13. The system of claim 5, wherein said template authoring tool is further adapted to create and modify a parameterized template having at least one global layout parameter, wherein said

parameterized template allows manipulation of said at least one global layout parameter of said parameterized template.

14. The system of claim 5, wherein said plurality of templates comprises an aggregate template that combines characteristics of a first template and a second template of said plurality of templates, wherein said first template and said second template define different aspects of document layout.
5
15. The system of claim 5, wherein said plurality of templates comprises an annotated template, wherein said annotated template includes a user-added attribute preference that assists said paginator in making layout and pagination decisions.
10
16. The system of claim 5, wherein said user interface is further adapted to create a guideline for a template of said plurality of templates, wherein said guideline is used to arrange elements within said template.
15
17. The system of claim 16, wherein said guideline has a position defined relative to the entire page of said template.
20
18. The system of claim 16, wherein said guideline is a first guideline and said user interface is further adapted to create a second guideline for said template of said plurality of templates, wherein said second guideline has a position defined as a constant offset from said first guideline, if said second guideline is created after selecting said first guideline.
25
19. The system of claim 16, wherein said guideline is a first guideline and said user interface is further adapted to create a second guideline for said template of said plurality of templates, wherein said second guideline has a position defined relative to said first guideline and a third guideline, if said second guideline is created after selecting said first guideline and said third guideline.
30

20. The system of claim 1, wherein said template comprises:
- at least one layout element that defines a particular region within a layout page of said template;
 - at least one constraint-based relationship that defines an arrangement of the layout elements within said template; and
 - at least one precondition that characterizes the suitability of said template for document content.
- 10 21. The system of claim 20, wherein said template storage unit is further adapted to store a plurality of adaptive layout styles.
- 15 22. The system of claim 21, wherein an adaptive layout style of said plurality of adaptive layout styles comprises a second plurality of templates and defines a layout design characteristic of said second plurality of templates.
- 20 23. The system of claim 20, wherein said at least one layout element comprises:
at least one source stream variable that specifies the type of document content which may be used in said at least one layout element.
24. The system of claim 20, wherein said at least one layout element comprises:
at least one element z-order placement variable that specifies the z-order within said template of said at least one layout element.
- 25 25. The system of claim 20, wherein said at least one layout element comprises:
at least one layout template variable that specifies a layout template used to layout a content item including at least one content stream.
- 30 26. The system of claim 20, wherein said at least one constraint-based relationship comprises:
at least one constraint input variable that includes data about the context of said document content; and

at least one constraint output variable that represents output attributes of said document content.

27. The system of claim 20, wherein said at least one precondition comprises:
 - 5 at least one content preconditions variable that indicates the amount of said document content required to fill said template.
28. The system of claim 20, wherein said at least one precondition comprises:
 - 10 at least one value preconditions variable that indicates a range of values for a predetermined constraint-based relationship.
29. The system of claim 27, wherein said at least one precondition characterizes the suitability of said template for a display context.
- 15 30. The system of claim 1, wherein said document content comprises a first content stream having a first type of content.
 - 20 31. The system of claim 30, wherein said document content further comprises a second content stream having a second type of content and wherein said first content stream comprises a first content item, said second content stream comprises a second content item, and said first content item references said second content item.
 - 25 32. The system of claim 31, wherein said first content stream comprises a plurality of content items including said first content item, and wherein said content items of said plurality of content items are placed on a page of a document with said second content item.
 - 30 33. The system of claim 30, wherein said first content stream comprises:
 - at least one content stream attribute that specifies how to evaluate document content within said first content stream.

34. The system of claim 30, wherein said first content stream comprises:
at least one style identifier that defines the styling of text within said first content stream.

5 35. The system of claim 30, wherein said document content comprises a third content stream having at least one content item including a plurality of content streams.

36. The system of claim 30, wherein said document content further comprises a second content stream including:

10 at least one content item including a plurality of versions of content.

37. The system of claim 36, wherein a selection of a version of content from said plurality of versions of content is dependent on whether inclusion of said version of content improves a score associated with how well said document content fits said template.

15 38. The system of claim 30, wherein said document content further comprises a second content stream including:
an optional content stream including at least one content item, wherein the selection of said at least one content item is dependent on whether inclusion of said at least one
20 content item improves a score associated with how well said document content fits said template.

39. The system of claim 38, wherein said score associated with how well said document content fits said template is dependent on at least a plurality of attribute values associated with said at least one content item of said optional content stream.

25 40. The system of claim 30, wherein said document content further comprises a second content stream and wherein said fifth content stream includes a content item adapted to express a content-dependent constraint.

30

41. A method of applying document content to adaptive templates, the method comprising the steps of:

selecting an appropriate set of templates to apply document content to;
determining the size and position of an element within a template of the set of
5 templates;
calculating two-dimensional regions of a layout page;
reducing the size of said two-dimensional regions according to element overlap
and z-ordering; and
inserting document content into said two-dimensional regions.

10 42. The method of claim 41, wherein the step of determining said size and position of an element within a template of the set of templates comprises propagating the size and position of each element within said template by utilizing a template constraint graph.

15 43. The method of claim 41, wherein the step of selecting the appropriate set of templates comprises the steps of:

analyzing content preconditions of each template against the document content;
and

20 evaluating value preconditions of each template against a plurality calculated constraint values respectively associated with each template.

44. The method of claim 43, wherein the step of selecting the appropriate set of templates further comprises the step of:

accepting a plurality of templates with successfully evaluated content
25 preconditions and value preconditions.

45. The method of claim 41, wherein the step of inserting document content into the two-dimensional regions comprises the steps of:

determining whether the document content is an image; and
30 if the document content is an image, performing a sequence of steps comprising:
scaling the image to fit a content region of the element; and

displaying the image inside the content region of the element.

46. The method of claim 41, wherein the step of inserting document content into the two-dimensional regions comprises the steps of:

5 determining whether the document content is text; and
 if the document content is text, performing a sequence of steps comprising:
 inserting the text into the content region of the element using a line-breaking algorithm.

10 47. The method of claim 46, wherein the step of inserting the text into the content region of the element using a line-breaking algorithm comprises inserting the text into the content region of the element using an optimal line-breaking algorithm.

15 48. The method of claim 41, wherein the step of inserting document content into the two-dimensional regions comprises the steps of:

 determining whether the document content is an inline figure; and
 if the document content is an inline figure, performing a sequence of steps comprising:
 placing the inline figure at a specified position in the content region of the
20 element, wherein the specified position is determined by the placement of previous content in the element.

25 49. The method of claim 48, wherein the step of performing the sequence of steps further comprises scaling the inline figure to fill the width of a column within the content region of the element.

50. The method of claim 45, the method further comprising the steps of:
 determining whether the document content comprises audio; and
 if the document content comprises audio, performing a second sequence of steps
30 comprising:

embedding a visual representation of the audio at a specified position in the content region of the element.

51. The method of claim 45, the method further comprising the steps of:

5 determining whether the document content comprises video; and

if the document content comprises video, performing a second sequence of steps comprising:

embedding the video at a specified position in the content region of the element.

10

52. A method of automatically resizing elements in a document, the method comprising the steps of:

determining whether the element is an image;

if the element is an image, performing a sequence of steps comprising:

15

setting at least one constraint variable of the element, wherein the at least one constraint variable includes dimension data of the image; and

computing an aspect ratio from the at least one constraint variable; and
altering dimensions of the element based on the aspect ratio.

20 53. The method of claim 52, wherein the step of setting at least one constraint variable of the element comprises setting the at least one constraint variable with pixel dimensions of the image.

25 54. The method of claim 53, wherein the step of computing an aspect ratio from the at least one constraint variable comprises computing the aspect ratio from the pixel dimensions.

55. The method of claim 54, wherein the step of altering dimensions of the element based on the aspect ratio comprises altering the height of the element to conform to the aspect ratio.

30

56. A method of automatically resizing elements in a document, the method comprising the steps of:
- determining whether the element is text;
 - if the element is text, performing a sequence of steps comprising:
 - 5 setting a dimension of the element to a predetermined value;
 - determining if a content region of the element is full; and
 - if the content region of the element is not completely filled by content, adjusting the dimension of the element to a second dimension.
- 10 57. The method of 56, wherein the step of setting a dimension of the element comprises setting a height variable of the element to a maximum allowable value.
- 15 58. The method of claim 57, wherein the step of adjusting the dimension of the element to a second dimension comprises adjusting the height variable of the element to the actual height of the text.
59. A method of automatically resizing elements in a document, the method comprising the steps of:
- determining whether the element is a compound element;
 - 20 if the element is a compound element, performing a sequence of steps comprising:
 - using a set of templates to layout the element, wherein a template of the set of templates includes a dimension variable having a value; and
 - setting a dimension of the element to the value of the dimension variable.
- 25 60. The method of 59, wherein the step of setting a dimension of the element comprises setting a height of the element to the value of the dimension variable.
61. A method for determining a sequence of templates for document layout, the method comprising the steps of:
- 30 laying out a portion of document content using a template;

- evaluating a template score variable of the template using the portion of document content;
- determining a number of widows and orphans in page layout resulting from application of the template to document content;
- 5 calculating a quality score of the template based on the template score variable and the determined number of widows and orphans; and
- using the quality score to influence a choice of a sequence of templates for paginating all document content.
- 10 62. A method of selecting a sequence of templates for optimally paginating document content, the method comprising the steps of:
- initializing a data table adapted to store a plurality of data entries, wherein a data entry of said plurality of data entries includes data representative of a location in a document layout and a sequence of templates that ends at the location;
- 15 calculating a quality score for the sequence of templates based upon template scores of individual templates in the sequence of templates and at least one global factor or at least one local factor;
- setting a threshold score, wherein the threshold score is used to determine whether a quality score for a sequence of templates is acceptable;
- 20 generating a new data entry, a quality score for sequence of templates ending with the new data entry, and a back pointer to a previous data entry of said plurality of data entries; and
- obtaining an optimal sequence of templates and content mapping by starting with the last entry of the data table and following a sequence of back pointers.
- 25
63. The method of claim 62, wherein said at least one global factor includes relative positions of a first content item of a first content stream and a second content item of a second content stream referenced by the first content item of the first content stream.
- 30 64. The method of claim 62, wherein said at least one local factor includes attributes assigned to the individual templates.

65. The method of claim 62, wherein the method further comprises the steps of:
generating a list of endpoints for a next page of document content starting at a
location;
- 5 calculating a quality score for each template sequence ending with each endpoint
of the list of endpoints;
- determining whether the numeric score of each template sequence is better than
the threshold score;
- if the numeric score of each template sequence is better than the threshold score,
- 10 performing a first sequence comprising the steps of:
determining whether a second data entry representing the location in the
document content of the endpoint already exists in the data table;
- generating a third data entry in the data table representing the location in
the document content of the endpoint;
- 15 storing the endpoint and a back pointer to the data entry in the data table,
if the second data entry does not already exist in the data table; and
- storing the endpoint and a back pointer to the data entry, if the quality
score is better than a stored quality score associated with the second data entry, if the endpoint
already exists in the data table.
- 20
66. The method of claim 65, wherein the step of calculating a numeric score comprises
evaluating a predetermined holistic metric for compositional balance of document content.
67. The method of claim 62, the method further comprising a step of modifying at least one
25 parameter of a template of the sequence of templates over a range of values.
68. The method of claim 62, wherein the step of setting a threshold score comprises applying
a “first-fit” pagination algorithm to the document content.

30

69. The method of claim 62, wherein the step of setting a threshold score comprises the steps of:

calculating the threshold score in a manner which ensures that the threshold score is better than a score generated from applying a “first-fit” pagination algorithm to the document content; and

adjusting the threshold score to be nearer the score generated from applying a “first-fit” pagination algorithm to the document content, if no valid sequence of templates for optimally paginating the document content is found.

10 70. A computer-readable medium having computer-executable instructions for applying document content to adaptive templates, the computer-executable instructions performing the steps of:

selecting an appropriate set of templates to apply document content to;

determining the size and position of an element within a template of the set of

15 templates;

calculating two-dimensional regions of a layout page;

reducing the size of said two-dimensional regions according to element overlap and z-ordering; and

inserting document content into said two-dimensional regions.

20

71. The computer-readable medium of claim 70, wherein the step of determining said size and position of an element within a template of the set of templates comprises propagating the size and position of each element within said template by utilizing a template constraint graph.

25 72. The computer-readable medium of claim 70, wherein the step of selecting the appropriate set of templates comprises the steps of:

analyzing content preconditions of each template against the document content;
and

evaluating value preconditions of each template against a plurality calculated
30 constraint values respectively associated with each template.

73. The computer-readable medium of claim 72, wherein the step of selecting the appropriate set of templates further comprises the step of:

accepting a plurality of templates with successfully evaluated content preconditions and value preconditions.

5

74. The computer-readable medium of claim 70, wherein the step of inserting document content into the two-dimensional regions comprises the steps of:

determining whether the document content is an image; and
if the document content is an image, performing a sequence of steps comprising:
10 scaling the image to fit a content region of the element; and
 displaying the image inside the content region of the element.

75. The computer-readable medium of claim 70, wherein the step of inserting document content into the two-dimensional regions comprises the steps of:

15 determining whether the document content is text; and
 if the document content is text, performing a sequence of steps comprising:
 inserting the text into the content region of the element using a line-breaking algorithm.

20 76. The computer-readable medium of claim 75, wherein the step of inserting the text into the content region of the element using a line-breaking algorithm comprises inserting the text into the content region of the element using an optimal line-breaking algorithm.

25 77. The computer-readable medium of claim 70, wherein the step of inserting document content into the two-dimensional regions comprises the steps of:

determining whether the document content is an inline figure; and
if the document content is an inline figure, performing a sequence of steps comprising:

30 placing the inline figure at a specified position in the content region of the element, wherein the specified position is determined by the placement of previous content in the element.

78. The computer-readable medium of claim 77, wherein the step of performing the sequence of steps further comprises scaling the inline figure to fill the width of a column within the content region of the element.

5

79. The computer-readable medium of claim 74, the computer-executable instructions further performing the steps of:

determining whether the document content comprises audio; and
if the document content comprises audio, performing a second sequence of steps

10 comprising:

embedding a visual representation of the audio at a specified position in the content region of the element.

80. The computer-readable medium of claim 74, the computer-executable instructions further 15 performing the steps of:

determining whether the document content comprises video; and
if the document content comprises video, performing a second sequence of steps comprising:

embedding the video at a specified position in the content region of the 20 element.

81. A computer-readable medium having computer-executable instructions for automatically resizing elements in a document, the computer-executable instructions performing the steps of:

determining whether the element is an image;
if the element is an image, performing a sequence of steps comprising:
setting at least one constraint variable of the element, wherein the at least one constraint variable includes dimension data of the image; and
computing an aspect ratio from the at least one constraint variable; and
altering dimensions of the element based on the aspect ratio.

30

82. The computer-readable medium of claim 81, wherein the step of setting at least one constraint variable of the element comprises setting the at least one constraint variable with pixel dimensions of the image.
- 5 83. The computer-readable medium of claim 82, wherein the step of computing an aspect ratio from the at least one constraint variable comprises computing the aspect ratio from the pixel dimensions.
- 10 84. The computer-readable medium of claim 83, wherein the step of altering dimensions of the element based on the aspect ratio comprises altering the height of the element to conform to the aspect ratio.
- 15 85. A computer-readable medium having computer-executable instructions for automatically resizing elements in a document, the computer-executable instructions performing the steps of:
 determining whether the element is text;
 if the element is text, performing a sequence of steps comprising:
 setting a dimension of the element to a predetermined value;
 determining if a content region of the element is full; and
 if the content region of the element is not completely filled by content,
20 adjusting the dimension of the element to a second dimension.
- 60 86. The computer-readable medium of 85, wherein the step of setting a dimension of the element comprises setting a height variable of the element to a maximum allowable value.
- 25 87. The computer-readable medium of claim 86, wherein the step of adjusting the dimension of the element to a second dimension comprises adjusting the height variable of the element to the actual height of the text.
- 30 88. A computer-readable medium having computer-executable instructions for automatically resizing elements in a document, the computer-executable instructions performing the steps of:
 determining whether the element is a compound element;

if the element is a compound element, performing a sequence of steps comprising:
 using a set of templates to layout the element, wherein a template of the
set of templates includes a dimension variable having a value; and
 setting a dimension of the element to the value of the dimension variable.

5

89. The computer-readable medium of 88, wherein the step of setting a dimension of the element comprises setting a height of the element to the value of the dimension variable.
90. A computer-readable medium having computer-executable instructions for determining a sequence of templates for document layout, the computer-executable instructions performing the steps of:
- 10 laying out a portion of document content using a template;
 - evaluating a template score variable of the template using the portion of document content;
 - 15 determining a number of widows and orphans in page layout resulting from application of the template to document content;
 - calculating a quality score of the template based on the template score variable and the determined number of widows and orphans; and
 - 20 using the quality score to influence a choice of a sequence of templates for paginating all document content.

91. A computer-readable medium having computer-executable instructions for selecting a sequence of templates for optimally paginate document content, the computer-executable instructions performing the steps of:
- 25 initializing a data table adapted to store a plurality of data entries, wherein a data entry of said plurality of data entries includes data representative of a location in a document layout and a sequence of templates that ends at the location;
 - calculating a quality score for the sequence of templates based upon template scores of individual templates in the sequence of templates and at least one global factor or at
 - 30 least one local factor;

- setting a threshold score, wherein the threshold score is used to determine whether a quality score for a sequence of templates is acceptable;
- generating a new data entry, a quality score for sequence of templates ending with the new data entry, and a back pointer to a previous data entry of said plurality of data entries;
- 5 and
- obtaining an optimal sequence of templates and content mapping by starting with the last entry of the data table and following a sequence of back pointers.

92. The computer-readable medium of claim 91, wherein said at least one global factor
10 includes relative positions of a first content item of a first content stream and a second content item of a second content stream referenced by the first content item of the first content stream.

93. The computer-readable medium of claim 91, wherein said at least one local factor includes attributes assigned to the individual templates.

- 15
94. The computer-readable medium of claim 91, wherein the computer-executable instructions further perform the steps of:
- generating a list of endpoints for a next page of document content starting at a location;
- 20 calculating a quality score for each template sequence ending with each endpoint of the list of endpoints;
- determining whether the numeric score of each template sequence is better than the threshold score;
- if the numeric score of each template sequence is better than the threshold score,
25 performing a first sequence comprising the steps of:
- determining whether a second data entry representing the location in the document content of the endpoint already exists in the data table;
- generating a third data entry in the data table representing the location in the document content of the endpoint;
- 30 storing the endpoint and a back pointer to the data entry in the data table, if the second data entry does not already exist in the data table; and

storing the endpoint and a back pointer to the data entry, if the quality score is better than a stored quality score associated with the second data entry, if the endpoint already exists in the data table.

5 95. The computer-readable medium of claim 94, wherein the step of calculating a numeric score comprises evaluating a predetermined holistic metric for compositional balance of document content.

10 96. The computer-readable medium of claim 91, wherein the computer-executable instructions further perform a step of modifying at least one parameter of a template of the sequence of templates over a range of values.

97. The computer-readable medium of claim 91, wherein the step of setting a threshold score comprises applying a “first-fit” pagination algorithm to the document content.

15 98. The computer-readable medium of claim 91, wherein the step of setting a threshold score comprises the steps of:

 calculating the threshold score in a manner which ensures that the threshold score is better than a score generated from applying a “first-fit” pagination algorithm to the document content; and

20 adjusting the threshold score to be nearer the score generated from applying a “first-fit” pagination algorithm to the document content, if no valid sequence of templates for optimally paginating the document content is found.

25